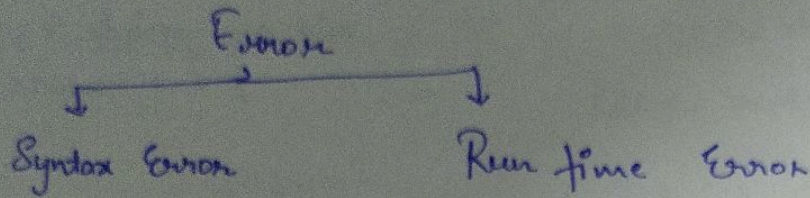


# # #  
# # #

## Exception Handling :



ax Error

= 10

x == 10

Print ("x value 10")

O/P! Syntax Error!

nt "Hello World"

O/P! Error for Python version 3

Not show error for ver-2  
because without () is valid

run Error: because of Invalid input  
Memory Problem

Wrong Programming logic

runtime Error called as Exception



Ex:

```
x = int(input("Enter first Number"))  
y = int(input("Enter Second Number"))  
print('The result:', x/y)
```

O/P:

Enter first Num: 4  
Enter Second Num: 2  
The result: 2

Suppose 1) we put  $y = 0$   
then Zero Division

Suppose we put  $x = 10$   
 $y = 0$   
Error: Value Error

Ex:

```
f = open('file.txt')  
print(f.read())
```

Error: File not found Error: # if file is not the

Exception handling concept is applicable only for Run  
Error only.



Ques 1: what is Exception?



Internet Error

1)



Sleeping Error

2)

g: am client meeting

8 am → 8:45

Timer Function Error

- 10:45

3)

An unwanted and unexpected event that disturbs normal flow of program is called Exception.

e.g. Zero Division Error  
Value Error

Ques 2: what is the main objective of Exception handling?

- Graceful termination / Normal Termination of the application  
(i.e. we should not block our resources and we should not miss anything)

4: → 8 PPT Re  
10: am

Ques 3: what is the Meaning of Exception handling?

Ex: Native Place by Bus

Report/PPT make  
Power go

but UPS 10

Power back

You store

PPT what

you can

do

Exception handling does not mean repairing an exception. we have to define alternative way to continue rest of the program normally.

The way of defining alternative is nothing but exception handling.



## : Raising Exception :

"Raise" built in keyword in Python

Raise builtin exception

Raise user defined exception

Finally

Exceptions in Python  
are instance of  
class

```
try :  
    raise MemoryError('Memory error')  
except MemoryError as e :  
    print(e)
```

↑ Builtin Exception

```
try :  
    raise Exception('memory error')  
except Exception as e :  
    print(e)
```

O/P: Memory Error

We are now creating user defined exception. Exception is based on instance/object of a class.

```
class Accident (Exception):  
    def __init__(self, msg):  
        self.msg = msg
```

```
def print_exception(self):  
    print("User defined exception: ", self.msg)
```

```
try:  
    raise Accident('Crash b/w two cars')  
except Accident as e:  
    e.print_exception()
```



Finally Statement :- always executed after try and except blocks.

```
def knows_file():
```

```
    try:
```

```
        f = open("c:\\code\\data.txt")
```

```
        x = 1/0
```

```
    except FileNotFoundError as e:
```

```
        print("inside except")
```

```
    finally:
```

```
        print("cleaning up file")
```

```
        f.close()
```

with this finally  
we are able to  
print this  
f.close() and  
after getting a  
Division Error.

O/P: cleaning up file

Zero Division Error

Syntax:

```
try:
```

```
    # Some code ...
```

```
except:
```

```
    # optional block
```

```
    # handling of exception (if required)
```

```
else:
```

```
    # executes if no exception
```

```
finally:
```

```
    # Some code ..... (always executed)
```



# Raising Exception in Python

raise keyword helps us to raise an exception  
Need to provide the name of exception class

try:

raise ZeroDivision

except Zero Division:

# Handling coding

## Try and Exception Example :-

```
def main():
```

```
    try:
```

```
        a = 1
```

```
        b = 1
```

```
        c = a/b
```

```
    except Exception as err:
```

```
        print("Error: {}".format(err))
```

```
    else:
```

```
        print("Did not Counter any exceptions --")
```

```
a = 1
b = 0
if b == 0:
    raise Exception("Zero Division Error")
    error has been raised
c = a/b
```